International Journal of Innovative
Computing, Information and Control
Volume **10**, Number **1**, February **2014**

 ISSN 1349-4198
pp. 303–313

TRANSDUCTIVE TRANSFER LEARNING BASED ON KL-DIVERGENCE

Jiana Meng[1,2] and Shichang Sun[2]

[1]School of Computer Science and Technology
Dalian University of Technology
No. 2, Linggong Road, Dalian 116023, P. R. China
mengjn@dlnu.edu.cn

[2]School of Computer Science and Engineering
Dalian Nationalities University
No. 18, Liaohexi Road, Dalian 116600, P. R. China
ssc@dlnu.edu.cn

Received January 2013; revised May 2013

Abstract. *Transfer learning solves the problem that the training data from a source domain and the test data from a target domain follow different distributions. In this paper, we take advantage of existing well labeled data and introduce them as sources into a novel transductive transfer learning framework. We first construct two feature mapping functions based on mutual information to re-weight the training and the test data. Then we compute the KL-divergence between the posterior probability of the unlabeled data and the prior probability of the labeled data to assign a pseudo-label to the unlabeled data. Next, a set of high-confidence newly-labeled data besides the labeled data are used for training a new classifier. The proposed algorithm requires that all unlabeled data in the target domain are available during training which is similar to the transductive learning setting, so we call it transductive transfer learning. The effectiveness of the proposed algorithm to transfer learning is verified by experiments in sentiment classification.*
Keywords: Transductive transfer learning, Sentiment classification, KL-divergence, Mutual information

1. **Introduction.** Traditional classification algorithms in machine learning assume that the training and the test data should share the same feature space and have the same data distribution. In real world applications, however, this assumption often does not hold. If there are very few labeled instances in the target domain for training, it is time-consuming to label them manually. So it would be favorable if we can leverage the labeled instances from the source domain to train a precise classifier for the target domain. In fact, the feasibility has been approved by frontier researches on transfer learning [1-3].

To transfer learning there are two classes of algorithms in the past, namely, feature-based algorithms and instance-based algorithms. For feature-based algorithms, Blitzer et al. [4] propose a structural correspondence learning (SCL) algorithm, which makes use of the unlabeled data from the target domain to extract some relevant features that may reduce the difference between domains. Dai et al. [5] propose a co-clustering based algorithm to propagate the label information across different domains. For instance-based algorithms, Taylor et al. [6] use source domain instances more selectively, and they use target domain instances to make decisions and only use source domain instances as insufficient target instances exist. Jiang and Zhai [7] propose a heuristic algorithm to remove "misleading" training instances from the source domain based on the difference between conditional probabilities. Meng et al. [8] propose an adaptive transfer learning

algorithm that adds the most similar instance to the training data set for solving the spam filtering problem.

Most of the above-mentioned transfer learning algorithms have achieved great improvement compared with traditional learning algorithms. However, the above-mentioned works do not consider how to combine the feature-based algorithm and the instance-based algorithm, which is the focus of this paper. We get the most similar features in the source domain to the features in the target domain based on mutual information and construct two feature mapping functions, which re-weight the source domain and the target domain data. We compute the KL-divergence between the posterior probability of the unlabeled data from the target domain and the prior probability of the training data from the source domain, which assign the label of the target domain data and add it to the source domain selectively. Finally, we retrain a new classifier, so that the hyper-plane can be revised to be closer to the distribution of the target domain data. Meanwhile, the proposed algorithm requires that all unlabeled data in the target domain are available at the training time, and is similar to the transductive learning setting, so we call it transductive transfer learning. We evaluate the proposed framework on the sentiment classification problem. Our experimental results show that the transfer framework significantly improves the performance over a number of baseline algorithms which shows the effectiveness of the proposed algorithm.

2. **Related Works.**

2.1. **Sentiment classification.** Automatic sentiment classification [9] is a supervised learning task. Although traditional classification algorithms [10] may be used to train sentiment classifiers from manually labeled data, however, the labeling work will be time-consuming and expensive. Meanwhile, if we directly apply a classifier trained in one domain to another domain, the accuracy performance will be very low due to the differences between the two domains. The reason is that users may use domain-specific words to express the sentiment in different domains. For instance, consider the simple case of training a system analyzing reviews about only two sorts of products: kitchen appliances and electronics. One set of reviews would contain adjectives such as "malfunctioning", "reliable" or "sturdy", and the other "sharp", "compact" or "blurry". Therefore, data distributions are different across domains. This violates the basic assumption of traditional supervised and semi-supervised classification algorithms. Recently, cross domain sentiment classification algorithms [11] have been proposed to solve the above problem.

2.2. **Transfer learning.** Another related learning research area is transfer learning [1-8,11]. It is an improvement of learning in a new domain through the transfer of knowledge from a related domain that has already been learned. At present many works have studied the cross domain sentiment classification, e.g., transfer learning is applied on solving sentiment classification tasks. Among those, a large majority of algorithms propose experiments performed on the benchmark made of reviews of Amazon products gathered by [11] which extends a structural correspondence learning algorithm [4] to sentiment classification. Besides sentiment classification, transfer learning algorithms have also been applied in many real world applications ranging from natural language processing [4,7], text categorization [12], visual concept detection [13] and WiFi localization [14].

2.3. **Transductive learning.** Transductive learning is an inference mechanism which uses both labeled data and unlabeled data to build a classifier whose main goal is that of classifying unlabeled data as accurately as possible. Traditional transductive learning setting assumes that the training data and the test data should follow the same data

distribution. Transductive SVM (TSVM) is a typical model employing testing data for transductive learning. Joachims [15] implements TSVM for text classification tasks with favorable results reported especially for problems with small training datasets.

3. **Transductive Transfer Learning Based on KL-Divergence.** The formulated problem in this paper is related to transfer learning, in which the major difficulty is that the source and the target domain data are not likely to be drawn from the same distribution. The intuitive solution seems to be simply trained on the target domain data. It has been shown that even small amounts of labeled target data can greatly improve transfer results [4], however, in this case no labeled target domain data are available. To solve this problem, a direct way is to make the unlabeled target test data be available to the model during training time. Leveraging the unlabeled test data during training time is called transductive learning. However, transduction is not well studied in a transfer setting, where the training and the test data come from different domains. To address the problem, we propose a transductive transfer learning algorithm based on the KL-divergence.

The proposed transfer learning algorithm is mainly composed of three steps. First, we obtain a common feature subset to both domains by using the feature-based algorithm. Then we learn two feature mapping functions to build a new vector space model (VSM) which ensures that the distributions of the training and the test data are close to each other. Finally, according to the KL-divergence between the posterior probability of the unlabeled data from the target domain and the prior probability of the labeled data from the source domain, we assign the label of the target domain data and add it into the source domain to retrain a new classifier, thus the instance-based algorithm is achieved.

3.1. **Notations.** To facilitate discussion, we introduce some notations. Let D_S be the set of the source domain data with labels and D_T be the set of the target domain data without labels. The feature sets of the source and the target domain data are F_S and F_T, respectively. F_S and F_T can be obtained from the feature occurrences in D_S and D_T. Here we denote $Y = \{0, 1\}$ by a set of class labels which is the corresponding output of the source domain data and the target domain data. Our target is to predict the class label corresponding to a data in D_T. Transfer learning aims to improve the learning of the target predictive accuracy score using the knowledge in D_S and D_T, where $D_S \neq D_T$.

3.2. **Feature subset.** In transfer learning, we have labeled data from a source domain, and we wish to learn a classifier which performs well on a target domain with a different distribution. Intuitively, if the two domains are related, there exist several common components underlying them. Some of these components may capture an intrinsic structure underlying the original data, while others may not. In order to discover those components, which do not cause the distribution change across the domains and capture the structure of the original data well, we need evaluate the interrelation of features between the source and the target domain. So we compute the co-occurrence features of the training and the test data, i.e., $F_C = F_S \cap F_T$. Those co-occurrence features are the representations of the interrelated degree of the two domains.

Next we select the seeds and the subject factors of the target domain. A seed is a target domain feature which does not belong to the source domain feature set. We denote the seeds by $\bar{F}_C$. Obviously, $\bar{F}_C = F_T - F_C$. Since the seeds belong to the target domain feature set and do not belong to the source domain feature set, the seeds can distinguish the target domain instances from the source instances explicitly. Then the subject factors are selected from the seeds. A subject factor is a high frequency feature of the seeds. We denote the subject factors by $\hat{F}$. Because of the high occurring frequency,

the subject factors are characteristic and important representation features. If we can find similar features between the subject factors and the features in F_C, the distribution distance of different domains data will be reduced. In the later experiment the accuracy of classification will be influenced by different appropriate threshold values of subject factors and seeds. Through the above-mentioned algorithm, we get a feature subset F', where F' is constructed by $\hat{F}$ and F_C, i.e., $F' = \hat{F} \cup F_C$.

3.3. **Feature interrelation perspective based on mutual information.** For the purpose of reducing the distribution distance of different domain data, we calculate the interrelated degree between the subject factors and the features of co-occurrence. The mutual information [16] algorithm is used to evaluate the similarity. The mutual information of two random variables expresses their mutual dependence or the amount of information they have in common. In other words, it measures how much knowing one of these variables reduces the uncertainty about the other. If the information shared between two objects is small, the two objects are likely to be independent. Otherwise, the two objects are unlikely to be independent of each other. Let w_i and w_j be two features, let $p(w_i)$ and $p(w_j)$ be the prior probability of w_i and w_j, respectively. The co-occurrence probability of w_i and w_j is $p(w_i, w_j)$. The mutual relationship between w_i and w_j is therefore $I(w_i, w_j)$, namely,

$$I(w_i, w_j) = \log \frac{p(w_i, w_j)}{p(w_i)p(w_j)} \tag{1}$$

The quantity $I(w_i, w_j)$ measures the mutual relationship between w_i and w_j or the probability that w_i and w_j make a joint contribution to a system at the same time. If they are statistically independent, the mutual information between them will be small. There are three possible values for $I(w_i, w_j)$: positive, zero and negative. A positive value means that w_i and w_j are statistically dependent: $I(w_i, w_j) > I(w_i)I(w_j)$. A negative value means that w_i and w_j can be regarded as complementary distribution because $I(w_i, w_j) < I(w_i)I(w_j)$, which states that the probability that w_i and w_j make a joint contribution to the system is less than the probability that w_i and w_j make separate contributions. A zero value means that w_i and w_j are statistically independent: $I(w_i, w_j) = I(w_i)I(w_j)$. The above analysis indicates that the mutual information of w_i and w_j is higher, the similarity of them is larger.

Assume that f_T^i is a subject factor and that f_S^j is a feature, $f_S^j \in F_C$, the mutual information of f_T^i and f_S^j can be defined as:

$$I(f_T^i, f_S^j) = \log \frac{p(f_T^i, f_S^j)}{p(f_T^i)p(f_S^j)} \tag{2}$$

where $p(f_T^i, f_S^j)$ is the feature frequency number that f_T^i and f_S^j co-occur, $p(f_T^i)$ is the feature frequency number that f_T^i occurs, $p(f_S^j)$ is the feature frequency number that f_S^j occurs.

The value of $I(f_T^i, f_S^j)$ is an interrelated measure between f_T^i and f_S^j. Assume that the value of $I(f_T^i, f_S^k)$ is the highest, then we can draw a conclusion that the most interrelated feature of f_T^i is f_S^k, namely,

$$sim(f_T^i) = f_S^j \tag{3}$$

3.4. **Feature mapping functions.** In Subsection 3.2, we build up a feature subset F'. On the basis of the feature subset we define two mapping functions $\zeta(D_S)$ and $\zeta(D_T)$. For a training set, if $d \in D_S$, the vector space model of d is $(f_S^1, f_S^2, \ldots, f_S^j, \ldots, f_S^k)$. In

the mapping $\zeta(D_S)$:

$$f_S^j = \begin{cases} 1 & \text{if } f_S^j \in F_C \text{ or } sim(f_T^i) = f_S^j(f_T^i \in \hat{F}) \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

Similarly, if $d \in D_T$, the vector space model of d is $(f_T^1, f_T^2, \ldots, f_T^i, \ldots, f_T^k)$. In the mapping $\zeta(D_T)$:

$$f_T^i = \begin{cases} 1 & \text{if } f_T^i \in F' \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

Through the mapping functions, we re-weight the source and the target domain data to reduce the distance of distributions between them. We build a new VSM to approximate the distribution of the target domain data.

3.5. **Prediction of the pseudo-label based on KL-divergence.** As mentioned above, the training data from a source domain follow different distributions with the test data from a target domain. For adjusting the distribution bias, we add the test data for retraining a new classifier, so that the hyper-plane can be revised to be closer to the distribution of the target domain. Figure 1 illustrates the revision procedure, in which Figure 1(a) shows the classifier trained on the labeled data from the source domain and Figure 1(b) shows the adjusted classifier after adding the test data with pseudo-label. The red diamonds show the test positive instances and the red circles show the test negative instances.

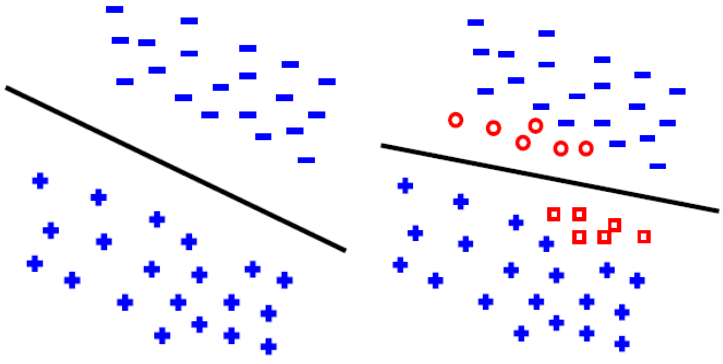
The pseudo-labels of the test data will be assigned based on the KL-divergence. Assume that θ_P is the true probability distribution of the positive instance in the source domain. θ_N is the true probability distribution of the negative instance in the target domain. For inferring the class label of a test data, we compute the distance between a test data and θ_P, and the distance between a test data and θ_N, respectively. Then the distribution function φ is defined as:

$$\varphi(d; \theta_P, \theta_N) = Dis(\theta_d, \theta_P) - Dis(\theta_d, \theta_N) \tag{6}$$

Assumed that $Dis(p, q)$ is the distance of distributions p and q. We use the KL-divergence to compute $Dis(p, q)$. The KL-divergence of probability distributions p and q, $L(p \,\|q)$ can be defined as:

$$L(p \,\|q) = \sum_x p(x) \log \left(\frac{p(x)}{q(x)} \right) \tag{7}$$

It is easy to show that $L(p \,\|q)$ is non-negative. $L(p \,\|q) = 0$ if and only if $p = q$. $L(p \,\|q)$ is not a true distance between two distributions since it is not symmetric and does not



(a) (b)

FIGURE 1. Classifier revised after adding the test data

satisfy the triangle inequality. Then the KL-divergence between θ_d and θ_P can be defined as:

$$L\left(\hat{\theta}_d \middle\| \hat{\theta}_P\right) = \sum_d \Pr\left(d \middle| \hat{\theta}_d\right) \log\left(\frac{\Pr\left(d \middle| \hat{\theta}_d\right)}{\Pr\left(d \middle| \hat{\theta}_P\right)}\right) \tag{8}$$

In a similar way, the KL-divergence between θ_d and θ_N can be defined as:

$$L\left(\hat{\theta}_d \middle\| \hat{\theta}_N\right) = \sum_d \Pr\left(d \middle| \hat{\theta}_d\right) \log\left(\frac{\Pr\left(d \middle| \hat{\theta}_d\right)}{\Pr\left(d \middle| \hat{\theta}_N\right)}\right) \tag{9}$$

$\hat{\theta}$ shows the estimated distribution model of true distribution model θ. On substitution of Equations (8) and (9) into Equation (6) yields

$$\varphi(d; \theta_P, \theta_N) = L\left(\hat{\theta}_d \middle\| \hat{\theta}_P\right) - L\left(\hat{\theta}_d \middle\| \hat{\theta}_N\right) = \sum_d \Pr\left(d \middle| \hat{\theta}_d\right) \log\left[\frac{\Pr\left(d \middle| \hat{\theta}_N\right)}{\Pr\left(d \middle| \hat{\theta}_P\right)}\right] \tag{10}$$

If $\varphi(d; \theta_P, \theta_N) < 0$, the label of d is positive, whereas the predicted label is negative. For each instance $d_j \in D_T$ $(j = 1, \ldots, n)$, according to $\varphi(d_j; \theta_P, \theta_N)$, the label of d_j can be assigned. We call the label pseudo. Then we select λ instances according to the value of the absolute of $\varphi(d; \theta_P, \theta_N)$ from higher to lower and add them to the training data set. The process that unlabeled instances having pseudo-labels are added to the training data set is the instance-based algorithm. Taking into account the two feature mapping functions constructed in the feature-based algorithm previously, we combine the feature-based and the instance-based algorithms in the transductive transfer learning strategy.

3.6. **Algorithm description.** The description of our algorithm is shown in Table 1. We first obtain a new feature subset. In Step 2, we select seeds and subject factors. Then the

TABLE 1. Algorithm of transductive transfer learning based on KL-divergence

Input: A labeled training data set D_S; An unlabeled test data set D_T ; A set Y of all the class labels. *Output: The label of instance d, $d \in D_T$.*
1 Preprocess the training and the test data, obtain F_S and F_T, and seek the intersection of F_S and F_T, $F_C = F_S \cap F_T$, construct the vector space models of the training and test data. *2 Repeat* *(1) Obtain the seeds and the subject factors of F_T according to the DF (instance frequency) from higher to lower. Get the new feature subset F'.* *(2) Compute the most interrelated feature of f_T^i ($f_T^i \in \hat{F}$), f_S^j ($f_S^j \in F_C$) by mutual information value of f_T^i, f_S^j and construct two new feature mapping functions $\zeta(D_S)$ and $\zeta(D_T)$.* *(3) Rebuild the vector space models of the training and the test data in the mapping functions $\zeta(D_S)$ and $\zeta(D_T)$.* *(4) Compute the KL-divergence and assign the pseudo-label of d_j.* *(5) Select and add some instances to the training data set. The number of the selected instances is determined by a parameter λ.* *Until the selected test data are not changed.* *3 Call a traditional machine algorithm to calculate the label of instance d ($d \in D_T$).*

subject factors are selected from higher to lower according to the feature frequency value. The most interrelated feature f_S^j ($f_S^j \in F_C$) of f_T^i ($f_T^i \in \hat{F}$) is obtained which measures the interrelated degree between features. We then construct two new feature mapping functions and vector space models of the training and the test data. Finally, we compute the KL-divergence and assign the pseudo-label of d_j. We select λ instances according to the KL-divergence and add them to the training data set. Iterating the above process until the selected test data is not changed. The final label predictions are decided by SVM.

4. **Experiments and Results.** To investigate these questions, we have conducted experiments using mutual information to compute interrelated features for rebuilding the new VSM and adding different numbers of target domain instances with pseudo-labels. The experimental results are presented in Tables 2, 3 and Figures 2, 3. The results are discussed in the corresponding sections.

4.1. **Data sets.** We evaluate our algorithm by one corpus, namely, Amazon review [11]. We use the Amazon reviews data set to select Amazon product reviews for four different product types: books, DVDs, electronics and kitchen appliances. This benchmark data set has been used in previous work [11] on cross domain sentiment classification. We can directly compare our algorithm against existing algorithms by evaluating on the Amazon reviews data set. The detail descriptions of the corpus are given in [11].

4.2. **Software and evaluation.** We evaluate the baseline classifier, i.e., Support Vector Machine, which has been proved to be effective on many machine learning tasks. Here, we use the Svm-light [17] with a linear kernel function, and the default values are assigned to the parameters. Finally, the accuracy is calculated as a performance evaluation. In our experiments, we compare the classification results based on the BOW (Bag-of-Word) representation of instances. BOW is the earliest approach used to represent the instance as a bag of words under VSM. Standard pre-processing is performed on the raw data. Stop words are eliminated, and stemming is unperformed with the data set.

4.3. **Comparison results.** To evaluate the benefit of our proposed algorithm, we compare the proposed algorithm with two baseline algorithms in Table 2. Next, we describe the algorithms.

• Baseline: This baseline simulates the effect of not performing the feature-based and the instance-based transfer learning algorithms. We simply train a binary classifier using SVM from the labeled reviews in the source domains and apply the trained classifier on a target domain. This can be considered as a lower bound that does not perform transfer learning.

• FBA (feature-base algorithm): FBA means we use the feature-based algorithm, however, do not use unlabeled data in the target domain to train a binary classifier.

• Proposed: This is the proposed algorithm described in this paper. We use the feature-based and the instance-based algorithms described in Section 3 to train a binary classifier.

As discussed previously, we apply our algorithm to one corpus. Table 2 shows the accuracy results. Column 1 describes the data sets of the corpus; Column 2 shows the baseline performances which is applied Svm-light to the data sets; Column 3 shows the performances obtained by FBA. It means that the models of the classifier are built on the feature subset F'. In the experiment the ratio of subject factors and seeds is 10%. The performances are improved compared with Column 2. This result indicates that using mutual information to compute interrelated features for rebuilding the new VSM can generate positive effect on accuracy. If the data from the source and the target domain

TABLE 2. Comparison of accuracy results of different data sets

Data sets	*Baseline*	*FBA*	*Proposed*
D vs B	*0.761*	*0.782*	***0.801***
E vs B	*0.680*	*0.736*	***0.766***
K vs B	*0.693*	*0.712*	***0.725***
B vs D	*0.782*	*0.795*	***0.811***
E vs D	*0.693*	*0.710*	***0.720***
K vs D	*0.716*	*0.778*	***0.806***
B vs E	*0.692*	*0.747*	***0.779***
D vs E	*0.714*	*0.732*	***0.749***
K vs E	*0.806*	*0.829*	***0.842***
B vs K	*0.723*	*0.785*	***0.805***
D vs K	*0.742*	*0.791*	***0.821***
E vs K	*0.831*	*0.848*	***0.855***
average	*0.736*	*0.770*	***0.790***

are closely related, the performance of the classifier is actually improved. In particular, FBA can still give stable results, and the accuracies are increased by 3.4% averagely. This demonstrates the robustness of FBA under feature-based transfer settings. The last column shows the performance of the proposed algorithm. Here, we see that the performance outperforms the baseline by 5.4 percentage points and FBA by 2 percentage points in the whole problems. This is most likely because, although the training and the test data are related, they are still drawn from different distributions and thus cannot be intermingled indiscriminately. However, through using the added test data based on the KL-divergence, the hyper-plane can be revised to be closer to the distribution of the target domain data which leads to the improving of the classifier performance.

4.4. **Parameter sensitivity.** There is one parameter λ which is the number that the added target instances with a pseudo-label in the proposed algorithm. In Figure 2, we show the accuracy under the different threshold values of λ. The x-axis is the value of λ, which represents the number of the test data which are selected and added to the training data set. The y-axis is the value of accuracy. The x-axis equals 0 means that no target unlabeled data are added into the training data set. As shown in Figure 2, adding some labeled target instances can greatly improve the performance for all data sets. Furthermore, with the increasing value of λ, the accuracy increases slightly. The reason is that the hyper-plane can be revised to be closer to the distribution of the target domain data after adding the test data. So any changes in classification accuracy can be directly attributed to the contribution of the unlabeled data. Another reason is due to the strong relativity between the transformed training data and the test data by usage of the feature-based algorithm.

4.5. **Comparison with previous works.** We compare our proposed algorithm with SCL and SCL-MI [11]. Next, we briefly describe those algorithms. SCL and SCL-MI are the structural correspondence learning (SCL) algorithms proposed by Blitzer et al. [11]. These algorithms utilize both labeled and unlabeled data in the benchmark dataset. SCL-MI selects pivots using the mutual information between a feature and a domain label. A point of difference between SCL and our proposed algorithm is that SCL allows a small number of labeled data from the target domain to learn the model from one domain to

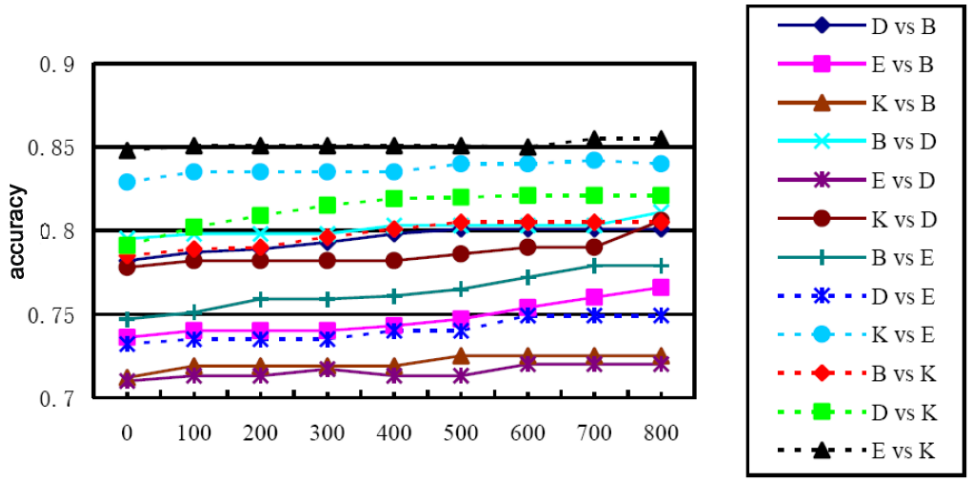


FIGURE 2. Parameter sensitivity of λ

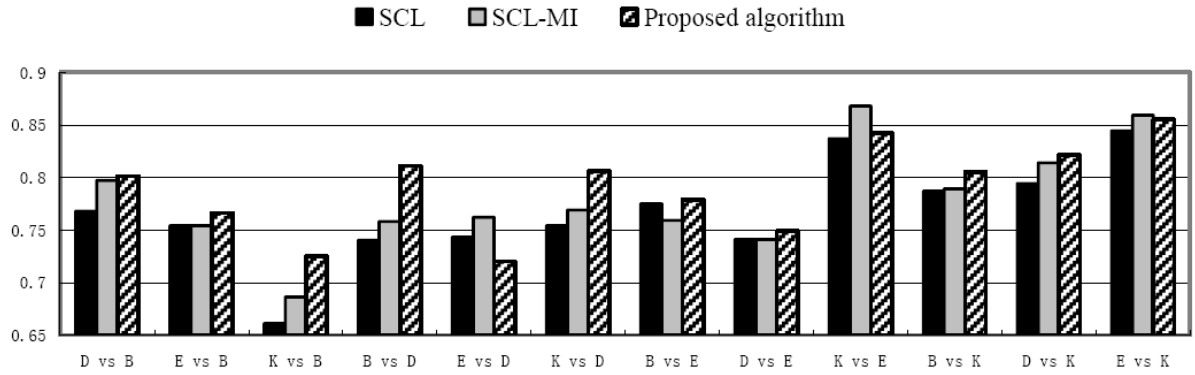


FIGURE 3. Accuracy results compared with SCL and SCL-MI

a new domain, however, our proposed algorithm does not use the labeled target domain data to train a binary sentiment classifier.

We first compare the proposed algorithm with previous works in Figure 3. Out of the 12 data sets compared in Figure 3, our proposed algorithm reports the best accuracies among all sentiment classification algorithms in 9 data sets except "E vs D", "K vs E", and "E vs K" data sets, whereas SCL-MI reports the best accuracies in the remaining 3 data sets. Obviously, our proposed algorithm performs uniformly better than SCL and SCL-MI in the most of the data sets.

Table 3 shows the average accuracy performances on the 4 target domains. As shown in Table 3, we see that the proposed algorithm outperforms SCL and SCL-MI in all target domains. Compared with the two algorithms, SCL and SCL-MI use the unlabeled target instances to infer a good feature representation, however, SCL and SCL-MI do not effectively use the unlabeled target instances to train a binary classifier. Furthermore, SCL and SCL-MI would not work if there were no labeled target instances. In some cases, even few of labeled target instances are scarce. In contrast, our proposed algorithm does not allow unlabeled target instances to contribute to the model estimation.

Furthermore, we can see the proposed algorithm further increases the classification performance. We believe the reason is that through the feature-based algorithm our approach can obtain the feature interrelation perspective; afterwards using through the instance-based algorithm our proposed algorithm can select more informative instances to retrain a new classifier. At the same time, those selected informative instances in the

Table 3. Comparison with previous works

	SCL	*SCL-MI*	*Proposed algorithm*
Books	0.728	0.746	***0.764***
DVDs	0.746	0.763	***0.779***
Electronics	0.784	0.789	***0.790***
Kitchen	0.808	0.821	***0.827***

target domain can effectively revise the original classifier because their distribution is similar to the distribution of the target domain data. Fusing the feature-based and the instance-based algorithms can improve the classification performance of their independent algorithm.

5. **Conclusions.** If the distribution of labeled data is different from that of unlabeled data, a classifier trained on labeled data can cause sub-optimal classification on unlabeled data. In this paper we propose a transductive transfer learning algorithm to solve this problem. We re-weight the source and the target domain data by computing the most similar features in the source domain to the features in the target domain. We compute the KL-divergence to bias class probabilities of labeled data and improve classification. We fuse the feature-based and the instance-based algorithms on sentiment classification task. The experimental results show that our algorithm can greatly outperform some existing state-of-the-art algorithms.

Acknowledgements. This work is supported by grants from the Natural Science Foundation of China (61202254), LiaoNing Provincial Education Department Project (L20124 78), China Postdoctoral Science Foundation Funded Project (2013M530918), and the Fundamental Research Funds for the Central Universities (DC120101084).

REFERENCES

[1] H. Daumé III and D. Marcu, Domain adaptation for statistical classifiers, *Journal of Artificial Intelligence Research*, vol.26, no.1, pp.101-126, 2006.

[2] C. Do and A. Ng, Transfer learning for text classification, *Advances in Neural Information Processing Systems*, vol.18, pp.299-306, 2006.

[3] R. Raina, A. Battle, H. Lee et al., Self-taught learning: Transfer learning from unlabeled data, *Proc. of the 24th Int. Conf. on Machine Learning*, pp.759-766, 2007.

[4] J. Blitzer, R. McDonald and F. Pereira, Domain adaptation with structural correspondence learning, *Proc. of the Conf. on Empirical Algorithms in Natural Language Processing*, pp.120-128, 2006.

[5] W. Y. Dai, G. R. Xue, Q. Yang et al., Co-clustering based classification for out-of-domain instances, *Proc. of the 13th Int. Conf. on Knowledge Discovery and Data Mining*, pp.210-219, 2007.

[6] M. Taylor, N. Jong and P. Stone, Transferring instances for model-based reinforcement learning, *Proc. of the European Conf. on Machine Learning*, pp.488-505, 2008.

[7] J. Jiang and C. Zhai, Instance weighting for domain adaptation in NLP, *Proc. of the 45th Annual Meeting of the Association of Computational Linguistics*, Prague, pp.264-271, 2007.

[8] J. N. Meng, H. F. Lin and Y. H. Yu, Adaptive transfer learning for spam filtering, *Journal of Computational Information Systems*, vol.6, no.13, pp.4581-4589, 2010.

[9] L. Yu, X. Liu, F. Ren and P. Jiang, Learning to classify semantic orientation on on-line instance, *International Journal of Innovative Computing, Information and Control*, vol.5, no.12(A), pp.4637-4645, 2009.

[10] B. Pang, L. Lee and S. Vaithyanathan, Thumbs up? Sentiment classification using machine learning techniques, *Proc. of the Conf. on Empirical Algorithms in Natural Language Processing*, pp.79-86, 2002.

[11] J. Blitzer, M. Dredze and F. Pereira, Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification, *Proc. of the 45th Annual Meeting of the Association for Computational Linguistics*, 2007.

[12] W. Y. Dai, G. R. Xue, Q. Yang et al., Transferring naive bayes classifiers for text classification, *Proc. of the 22nd AAAI Conference on Artificial Intelligence*, pp.540-545, 2007.

[13] L. X. Duan, I. W. Tsang, D. Xu et al., Domain transfer svm for video concept detection, *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

[14] S. J. Pan, I. W. Tsang, J. T. Kwok et al., Domain adaptation via transfer component analysis, *Proc. of the 21st International Joint Conference on Artificial Intelligence*, pp.1187-1192, 2009.

[15] T. Joachims, Transductive inference for text classification using support vector machines, *Proc. of the 16th Int. Conf. on Machine Learning*, pp.200-209, 1999.

[16] Z. R. Yang and M. Zwolinski, Mutual information theory for adaptive mixture models, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol.23, no.4, pp.396-403, 2001.

[17] T. Joachims, *Learning to Classify Text Using Support Vector Machines*, Ph.D. Thesis, 2002.